WriteScad

By Harlan Martin
harlan@sutlog.com

January 2012

Version 2

- Added support for future font selection (default is Letters.dxf)
- Added WriteCube module
- Added Rotate for text (rotates on the plane of the text)
- Added writesphere module

NOTE: These routines require openscad version 2011.12 or later

http://www.openscad.org/

-o-

write()

Note about strings:

1... The quote " symbol cannot be inserted into a string alone
Either use \" or just the single quote' will show as "

2... The \ is used for special text characters, so use the bar | to
show the back slash \

Usage

Put the files (**write.scad**) and (**letters.dfx**) in the
same directory as your project.

write("Hello World!",t=3,h=5,center=true);

(**t=**mm) (optional) The thickness of the letters in mm.
The default is 1mm if not specified

(**h=**mm)(optional) The height of the letters in mm.
The default is 4mm if not specified

(**rotate=**degrees) Rotates the text along the plane it is written on.

(**center=**boolean) (optional) Centers the text at default coordinates.

(**space=**character widths) (optional) Resizes distance between letters.
space=1 is default.. space=.8 will squeeze characters closer together.

(**font=**"font.dxf") (optional) Selects font. The selected font will need to be in
the working folder. So far, I have added "letters.dxf" (default)
"braille.dxf" "blackrose.dxf" "orbitron.dxf" "knewave.dxf"
(change the font="letters.dxf" in write.scad to change the default font)

**Note: These options apply to all modules including
write() writecube() writesphere() ect..**

Examples:

```
use <write.scad> // Dont forget to include this line

//example1: Uses all declarations..
        translate([20,15,0])
        write("Example 1",t=4,h=5.75,center=true);

//example2: Quick and easy
        write("That was easy!",h=12);

//example3: move and rotate(front) (remember to translate..then rotate)
        translate([0,0,10])
        rotate(90,[1,0,0])  // rotate around the x axis
        write("Rotate +X 90 (front)",t=2);
```

```
//example4: move and rotate(left side)
        translate([0,0,20])
        rotate(90,[1,0,0])   // rotate around the x axis
        rotate(90,[0,-1,0])  // rotate around the y axis
        write("Rotate +X 90 and -Y 90 (left side)");

//example5: move and rotate(right side)
        translate([0,0,30])
        rotate(90,[1,0,0])   // rotate around the x axis
        rotate(90,[0,1,0])  // rotate around the y axis
        write("Rotate +X 90 and +Y 90 (right side)");

//example6: move and rotate(back)
        translate([0,0,40])
        rotate(90,[1,0,0])   // rotate around the x axis
        rotate(180,[0,1,0])  // rotate around the y axis
        write("Rotate +X 90 and +Y 180 (back)");

//These examples are contained in TestWrite.scad
```

-o-

writecube()

For these examples, assume we have the cube:

```
use <write.scad>
translate([10,20,30])
cube(30,center=true);
```

text where and size

The values for **text=**, **where=** and **size=** are required, but if the values are entered in this order, then the commands are not required. These three examples produce the same results.

```
writecube(text="text", where=[10,20,30], size=[30,30,30]);
```

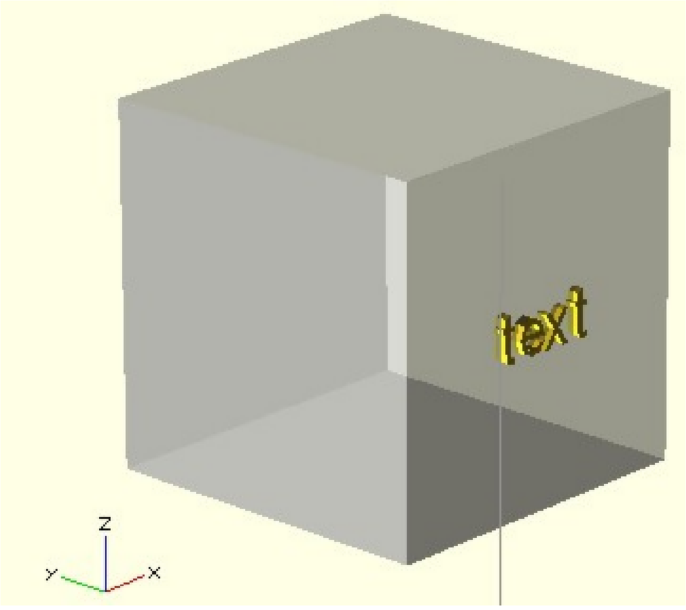
or

```
writecube("text",[10,20,30], [30,30,30]);
```

or

```
writecube("text",[10,20,30],30);
```

text is written on the center front of a cube that is **30**mm on all sides, and is centered at **[10,20,30]**



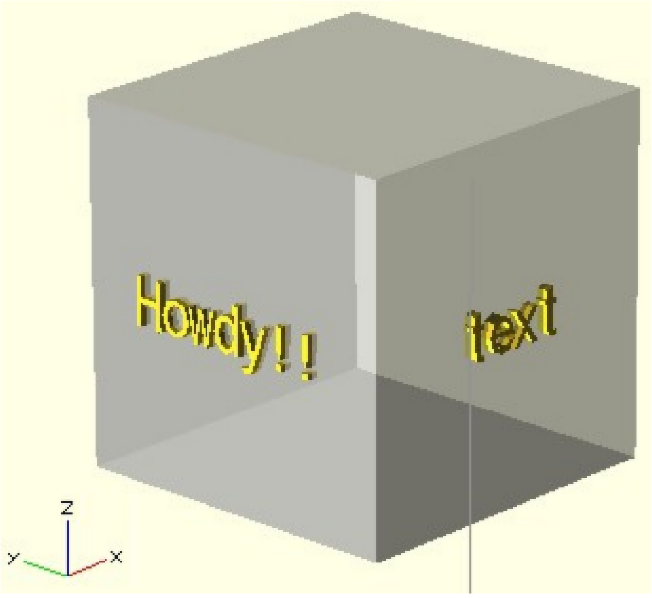
text="whatever text you want to write"
where= the center coordinates of the box
size = size of cube. If the cube is not square, use the format **[xsize,ysize,zsize]**
If it is the same on all sides, just give the size *ie.* **30**

face=

By default, **writecube()** will write on the front face of the box.
This assumes that x=left to right, y=front to back, z=bottom to top.
To write on the other sides, use:
face="top",**face="bottom"**,**face="back"**,**face="front"**,
face="left" or face="right"

```
writecube("Howdy!!",[10,20,30],30,face="left");
```

will print **Howdy!!** on the center left of the box.

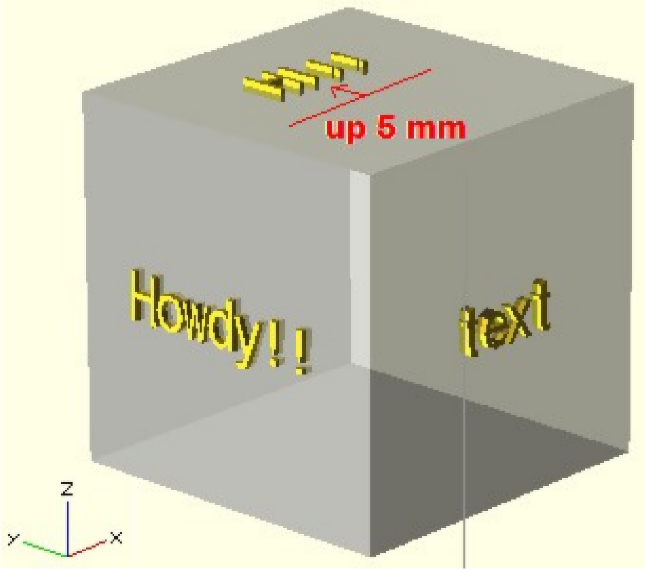


left right up and down

If you dont want the text centered, use left=mm or up=mm or down..right..
These commands move the text along the plane in the givin
direction **(in relation to the unrotated text)** in milimeters.

writecube("HI!!",[10,20,30],30,face="top",up=5);

will write **HI!!** 5mm up from the center along the top plane of the box.



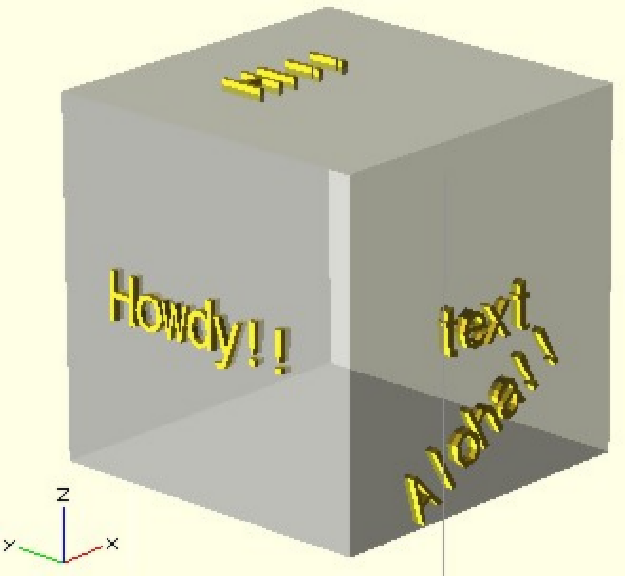
(Note: up down left right refer to their 2 dimensional counterparts here)

rotate

You say you dont want the text parallel with the sides?
rotate = will fix that for you. It rotates the text clockwise
along the plane of the text. (in degrees)

writecube("Aloha!!",[10,20,30],30,face="front",down=8,rotate=-30);

wil rotate **Aloha!!** counter-clockwise 30 degrees on the front of the box.



text size and thickness

t=how thick the text will be in mm
h=height of the font or fontsize
if not specified, the text will be 4mm tall (upper case)
and 1mm thick. (half inside and half outside the cube)

```
writecube("Hello!!",[10,20,30],30,face="right", t=2,h=4);
```

will write **Hello!!** on the right side of the cube with 1mm sticking out.

Keep in mind,half the thickness of the text will be outside, half inside. This makes it easy to create indented or protruding text on your designs.

Note: All options from write() apply to writecube()

-o-

writesphere

For these examples, assume we have the sphere:

```
use <write.scad>
translate([0,0,0])
sphere(10);
```

text where and radius

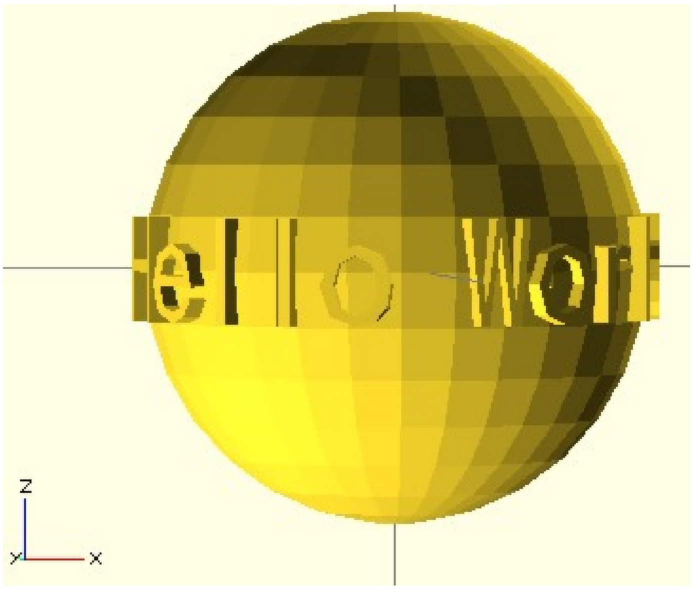
The values for **text=**, **where=** and **radius=** are required, but if the values are entered in this order, then the commands are not required. These two examples produce the same results.

writesphere(text="Hello World", where=[0,0,0], radius=10);

or

writesphere("Hello World",[0,0,0], 10);

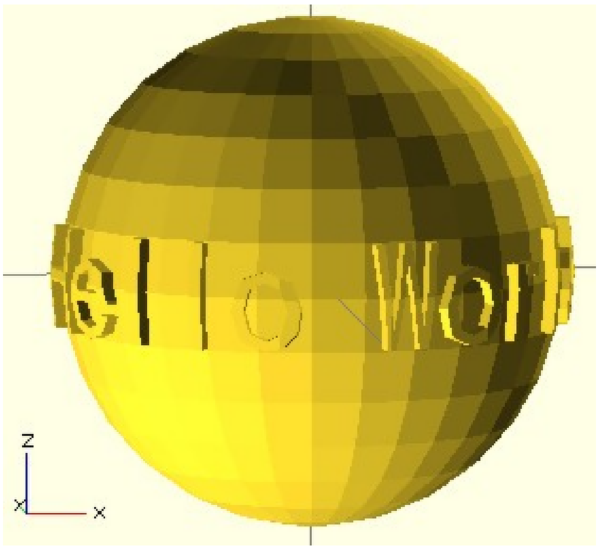
Hello World is written on the center front of the sphere.



Rounded

rounded= true or false (default = false)
If the text is very large compaired to the sphere, the flat text might not conform to the sphere. Either make the text thicker, or make it rounded. Rounded text takes a lot longer to render, so be patient. I suggest placing the text and only rounding it when the model is finished. (NOTE: $fn= works here too)

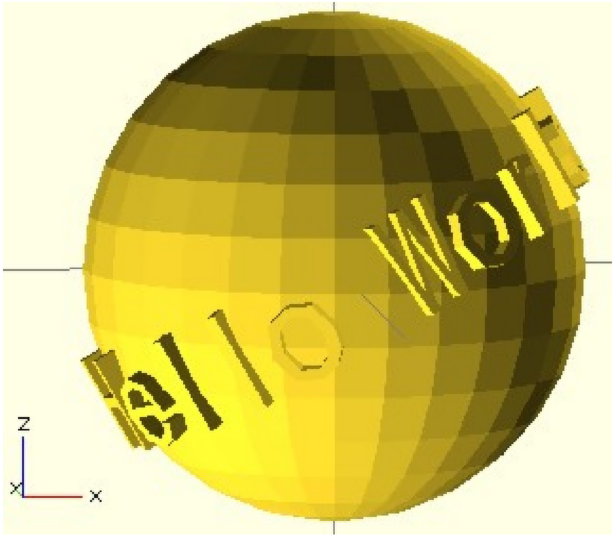
writesphere("Hello World",[0,0,0],10,rounded=true);



Spin

spin=degrees
writesphere("Hello World",[0,0,0],10,spin=-30);

Spins **Hello World** 30 degrees counter-clockwise

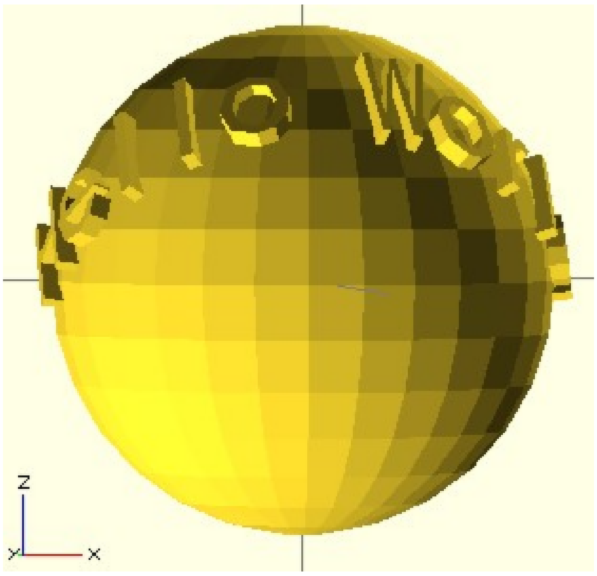


north and south

north=degrees or **south=**degrees will rotate the center of the text north or south.

writesphere("Hello World",[0,0,0],10,north=45);

Hello World is rotated north 45 degrees

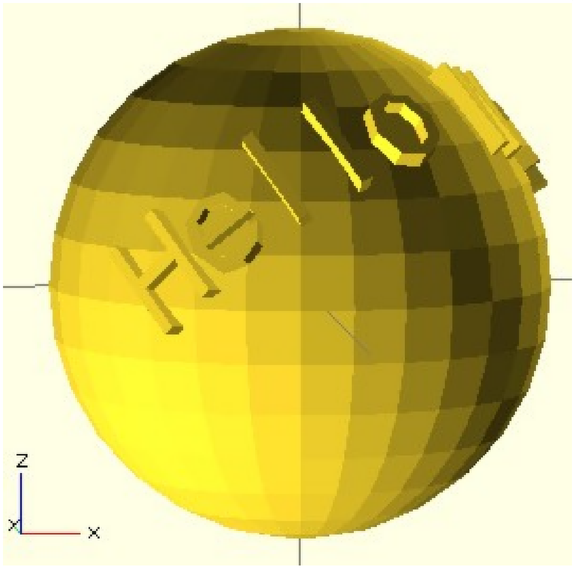


east and west

east=degrees or **west=**degrees will rotate the center of the text east or west

writesphere("Hello World",[0,0,0],10,north=45,east=45);

Hello World is rotated north 45 degrees and east 45 degrees



Note: All options from write() apply to writecube()

-o-